

A continuación introduciremos el concepto de hipercategoría y, siguiendo el camino trazado en el capítulo anterior, veremos los lenguajes que generan estos objetos para luego notar que hemos avanzado en la jerarquía de Chomsky, ya que estos son precisamente los lenguajes libres de contexto, o de tipo 0. Finalmente, implementaremos estos conceptos en `Python`.

0.1. Hipercategorías

Definición 1. Una **signatura de hipercategoría** es un conjunto de nodos H_1 y uno de objetos H_0 junto a un par de funciones

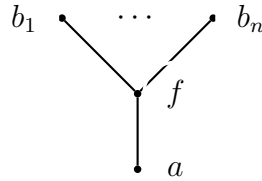
$$H_0^* \xleftarrow{\text{dom}} H_1 \xrightarrow{\text{cod}} H_0$$

Un morfismo de signaturas de hipercategorías $\varphi : H \rightarrow \Gamma$ es un par de funciones $\varphi_0 : H_0 \rightarrow \Gamma_0$ y $\varphi_1 : H_1 \rightarrow \Gamma_1$ tal que el siguiente diagrama conmuta:

$$\begin{array}{ccccc} H_0^* & \xleftarrow{\text{cod}} & H_1 & \xrightarrow{\text{dom}} & H_1 \\ \downarrow \varphi_0 & & \downarrow \varphi_1 & & \downarrow \varphi_0 \\ \Gamma_0^* & \xleftarrow{\text{cod}} & \Gamma_1 & \xrightarrow{\text{dom}} & \Gamma_1 \end{array}$$

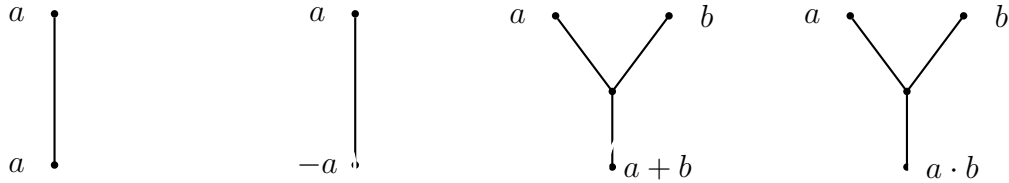
Con tales morfismos, las signaturas de hipergráficas forman la categoría **HypCat**.

Dado una signatura de hipergráfica, denotaremos un nodo $b_1 \dots b_n \xrightarrow{f} a$ graficamente como:



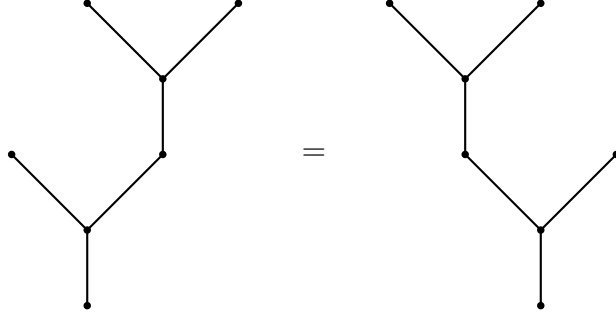
Además, diremos que f tiene aridad n .

Ejemplo 1. Consideremos la siguiente signatura de hipergráfica donde los objetos corresponderán a los números reales, \mathbb{R} . Luego, dado $a, ab \in \mathbb{R}^*$ tenemos los siguientes nodos:



Notemos que los primeros dos nodos corresponden a las operaciones de aridad 1 que lleva un número así mismo o a su inverso aditivo, respectivamente. Por otro lado, los últimos dos nodos corresponden a las operaciones binarias usuales de los números reales: la adición y la multiplicación.

Notemos, además, que para cualquier $abc \in \mathbb{R}^*$ tenemos que, dado que nuestras operaciones son asociativas, entonces $a + (b + c) = (a + b) + c$, así gráficamente tenemos la igualdad de los siguientes diagramas:



Por ello, siguiendo un razonamiento inductivo para cualesquiera $a_1 \dots a_n \in \mathbb{R}$ podemos agregar los nodos:



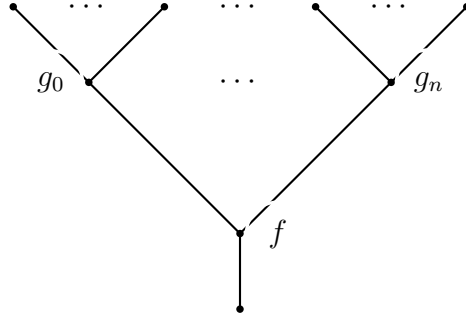
Así, todos los nodos son los representados por alguno de los diagramas anteriores. Con lo anterior, tenemos una signatura de hipercategoría.

Ahora, veamos la definición principal de esta sección:

Definición 2. Un **hipercategoría** es una signatura de hipercategoría \mathbf{H} junto a una operación de composición $\Pi_{b_i \in \vec{b}} \mathbf{Hyp}(\vec{c}_i, b_i) \times \mathbf{Hyp}(\vec{b}, a) \rightarrow \mathbf{Hyp}(\vec{c}, a)$ con $a \in \mathbf{O}_0$ y $\vec{c}, \vec{b} \in \mathbf{O}_0^*$. Dadas $f : \vec{b} \rightarrow a$ y $g_i : \vec{c}_i \rightarrow b_i$ definimos la composición como:

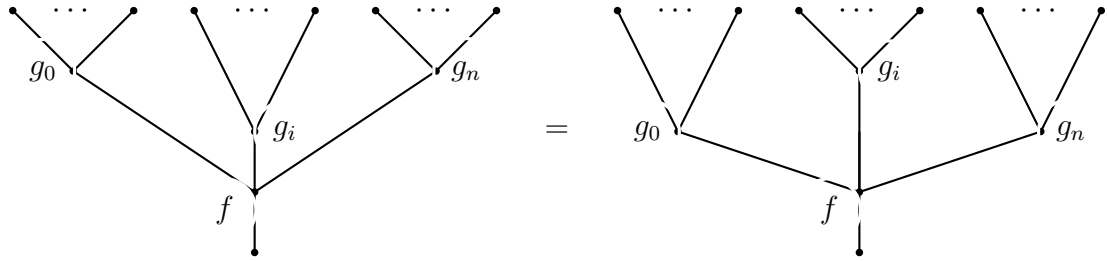
$$(c_1^1, c_2^1, \dots, c_{m_1}^1, c_2^1, \dots, c_1^n, \dots, c_{m_n}^n) \mapsto f(g_1(c_1^1, \dots, c_{m_1}^1), \dots, g_n(c_1^n, \dots, c_{m_n}^n))$$

que representamos gráficamente de la siguiente manera



Además, para cualquier $a \in \mathbf{H}_0$ tenemos una identidad $a \xrightarrow{Id_a} a$ en donde se satisfacen las siguientes propiedades:

1. Para cualquier $f : \vec{b} \rightarrow a$ en \mathbf{O}_1 , $f \cdot Id_a = f = \vec{Id}_{b_i} \cdot f$; y
2. Para cualesquiera $f : \vec{b} \rightarrow a$ y $g_i : \vec{c}_i \rightarrow b_i$ tenemos que:



Un álgebra $F : \mathbf{H} \rightarrow \mathbf{N}$ es un morfismo de signature de hipercategorías compatible con la composición, es decir, cada que $\vec{c} \xrightarrow{\vec{g}} \vec{b} \xrightarrow{f} a$, $F(\vec{g} \cdot f) = F(\vec{g}) \cdot F(f)$

Es importante resaltar que la propiedad 2 nos indica que no importa el orden en que hagamos las composiciones, es decir, se respeta cierta noción de asociatividad. Vale la pena resaltar que todavía no definimos formalmente el significado del cálculo de diagramas presentado, pues lo analizaremos con detenimiento en el siguiente capítulo.

Notemos que la signature de hipercategoría del ejemplo anterior es propiamente un hipercategoría con la composición dada de la manera obvia.

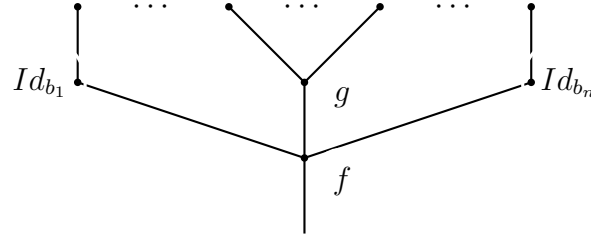
De manera análoga al capítulo anterior, nuestra intención es definir una hipercategoría libre. Antes, veamos la siguiente definición:

Definición 3. Sea \mathbf{H} una hipergráfica. Sea $f : \vec{b} \rightarrow a$ y $g : \vec{c} \rightarrow b_i$ en \mathbf{H}_1 con $i \in \{1, \dots, n\}$ con n es la aridad de f .

Definimos la composición parcial en la i -ésima entrada como:

$$g \circ_i f = (id_{b_1} \cdots g \cdots id_{b_n}) \cdot f$$

Graficamente



Una observación importante es que definimos la composición parcial a partir de la composición, pero también pudimos definirlo inversamente de la siguiente forma:

$$[g_1 \cdots g_n] \cdot f = g_n \circ_n (g_{n-1} \circ_{n-1} \cdots \circ_2 (g_1 \circ_1 f) \cdots)$$

Notemos que la propiedad 2 de la composición para hipergráficas nos garantiza que efectivamente la composición anterior coincide con la usual.

Con ello, ya estamos casi en condiciones de mostrar la construcción de un hipercategoría libre, pero antes veamos una definición más:

Definición 4. Sea H una signature de hipercategoría, definimos recursivamente un árbol con raíz a de H , como:

- Un nodo de la forma $w : \varepsilon \rightarrow a$ es un árbol (a los nodos de esta forma, les llamamos hojas);
- Un par $(f, (g_1, \dots, g_n))$ donde f es un nodo de aridad n , $f : a_1 a_2 \cdots a_n \rightarrow a$, y g_1, \dots, g_n son nodos con codominios a_1, \dots, a_n es un árbol;
- Son todos.

Sea H una signature de operad. $\mathbf{Hyp}(H)$ es la hipercategoría cuyos objetos son exactamente los mismos que G con los morfismos los árboles en G y para cada objeto $a \in G_0$ tenemos la id_a . La i -ésima composición parcial para morfismos f, g en $\mathbf{Op}(G)$, $g \circ_i f$ está por reemplazar la i -ésima hoja de f por una copia de g . **La definición puede ser mas rigurosa recursivamente.** Decimos que dos árboles en $\mathbf{Op}(G)$ son iguales si puede ser obtenido uno a partir de otro mediante la aplicación consecutiva de las propiedades 1 y 2 para la composición en operads.

Es claro que la composición parcial anterior induce correctamente una composición de operads y, por lo tanto, $\mathbf{Op}(G)$ efectivamente es un operad.

Más aún, la construcción anterior es libre en virtud de la siguiente proposición:

Proposición 1. Sean G una signature de operad, \mathbf{O} un operad y $f : G_1 \rightarrow O_1$ una función que preserve la aridad de los nodos. Entonces existe un único morfismo de operads $\varphi : \mathbf{Op}(G) \rightarrow \mathbf{O}$ tal que el siguiente diagrama conmuta:

$$\begin{array}{ccc} G & \xrightarrow{\eta} & \mathbf{O} \\ \downarrow i & \searrow \varphi & \uparrow \\ \mathbf{Op}(G) & & \end{array}$$

Resta decir que $\mathbf{Op} : \mathbf{OpSig} \rightarrow \mathbf{Operad}$ es un funtor y, con ello, tenemos la última proposición de nuestra sección:

Proposición 2. *La construcción de un operad libre es parte de la siguiente adjunción:*

$$\mathbf{OpSig} \begin{array}{c} \xrightarrow{\mathbf{Op}} \\ \xleftarrow{U} \end{array} \mathbf{Operad}$$

donde U es el funtor olvidadizo de un operad a su signature.

0.2. Lenguajes libres de contexto

Sea G signature finita de operad. Consideremos sus objetos como símbolos, los nodos como reglas de producción y los morfismos en $\mathbf{Op}(G)$ como derivaciones, entonces obtenemos la noción de una gramática, más precisamente:

Definición 5. Una gramática operádica como una signature finita de operad de la siguiente forma:

$$(B + V)^* \leftarrow G \rightarrow B$$

donde V es un vocabulario y B es un conjunto de símbolos no terminales con un símbolo distinguido $s \in B$ y G un conjunto de reglas de producciones.

El lenguaje generado por G es:

$$\mathcal{L}(G) = \{u \in V^* \mid \exists g : u \rightarrow s \in \mathbf{Op}(G)\}$$

Agregar dos ejemplo. Lógica proposicional y del lenguaje

Ahora, como anunciamos al inicio del capítulo, mostraremos la relación entre los lenguajes anteriores con la jerarquía de Chomsky. Para ello, veamos la siguientes definiciones y resultados.

Definición 6. Una gramática libre de contexto es una tupla $G = (V, B, P, s)$ donde V es un vocabulario, B es un conjunto de símbolos no terminales con un símbolo distinguido s y P es un conjunto de reglas de producción de la forma $A \rightarrow \alpha$ donde A es un símbolo no terminal y $\alpha \in (V + B)^*$.

Si $A \rightarrow \alpha$ es una regla de producción, para cualquier $\beta, \gamma \in (V + B)^*$ definimos la derivación $\beta A \gamma \Rightarrow \beta \alpha \gamma$. Consideramos \Rightarrow^* como la cerradura transitiva de \Rightarrow .

Definimos el lenguaje generado por G como:

$$\mathcal{L}(G) = \{w \in V^* \mid \exists s \Rightarrow w\}$$

Decimos que un lenguaje L es libre de contexto si existe una gramática G libre de contexto tal que $L = \mathcal{L}(G)$

Observemos que una oración en un lenguaje libre de contexto corresponde a una cadena en V^* que puede ser derivada a partir de s en G .

Ahora, la manera anterior de describir el lenguaje es eficiente para la generación de nuevas expresiones, sin embargo, puede no ser muy útil para el propósito de discriminar si una expresión dada pertenece, o no, al lenguaje. Con ese propósito surgen los árboles de derivación.

Definición 7. Sea $G = (V, B, P, s)$ una gramática libre de contexto.

Un árbol es un árbol de derivación para G si:

- Cada vértice está etiquetado con un símbolo de $V \cup B \cup \{\varepsilon\}$;
- La raíz del árbol es s ;
- Si un vértice interior está etiquetado con A , entonces $A \in B$, es decir, no es un símbolo terminal.
- Si un vértice A tiene n hijos, A_1, \dots, A_n , entonces:

$$A \rightarrow A_1 A_2 \cdots A_n$$

es una regla de producción en P .

- Si un vértice está etiquetado con ε , entonces es un hoja y, además, es el único hijo de su padre.

Notemos que el diagrama del ejemplo uno corresponde a un árbol de derivación en la gramática $A \rightarrow \neg A | A \wedge A | A \vee A | A \rightarrow A | A \leftrightarrow A | a_0 | a_1 | a_2$.

Con esto, podemos enunciar la siguiente proposición cuya prueba podemos consultar en [Agregar ref].

Proposición 3. Sea $G = (V, B, P, s)$ una gramática libre de contexto. Entonces $s \Rightarrow^* w$ si y sólo si existe un árbol de derivación para G que produce a w .

Es fácil notar que un morfismo en un operad corresponde, de hecho, a un árbol de derivación. Con ello, tenemos el siguiente corolario:

Teorema 1. Sea G una signature de operads y G' una gramática libre de contexto, entonces

1. $L(G)$ es un lenguaje libre de contexto.
2. Existe una signature de operad que genera a $L(G')$.

Demostración. **Pendiente.**

□